



SOLID - Open/Closed

Programação Orientada a Objeto II

SOLID - Open-Closed

Programação Orientada a Objeto II

"Objetos ou entidades devem ser abertos para extensão, mas fechados para modificações".

Sabe quando você define uma classe, mas parece que sempre surge algum novo atributo ou método para adicionar? Ou naquele método, parece que sempre aparece um novo if para adicionar? **Isto é uma violação do princípio Aberto-Fechado.**

O principal risco de violar esse princípio é criar novos bugs em um código ou funcionalidade que já existiam anteriormente - e, espera-se, funcionavam bem.

Nesse princípio é dito que, se você precisa fazer alterações no seu código-fonte, ele estará **feriando** o conceito de aberto e fechado. Portanto, uma classe deve estar aberta para que sejam feitas extensões nela. Deve ser possível a adição de código, mas não a modificação.

Problemática

Considere, novamente, nossa aplicação de secretaria de uma escola, mas, agora, a representação para um professor

```
public class Professor {  
    private String nome;  
}  
  
public class PagamentoService {  
    public void pagar(Professor  
professor) {  
        //code  
    }  
}
```


SOLID - Open-Closed

Programação Orientada a Objeto II

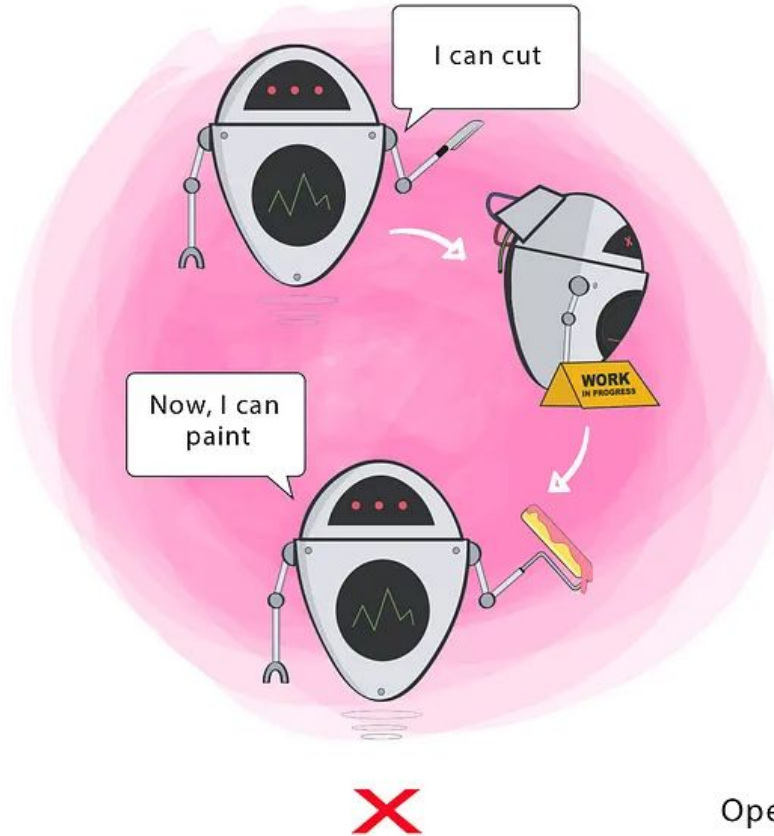
Solução

Segundo o princípio Aberto-Fechado, devemos herdar a classe Professor criando um novo tipo `ProfessorEmTreinamento` e, também, herdar cada classe onde as verificações deveriam ser feitas, sobrescrevendo os métodos com comportamento diferente.

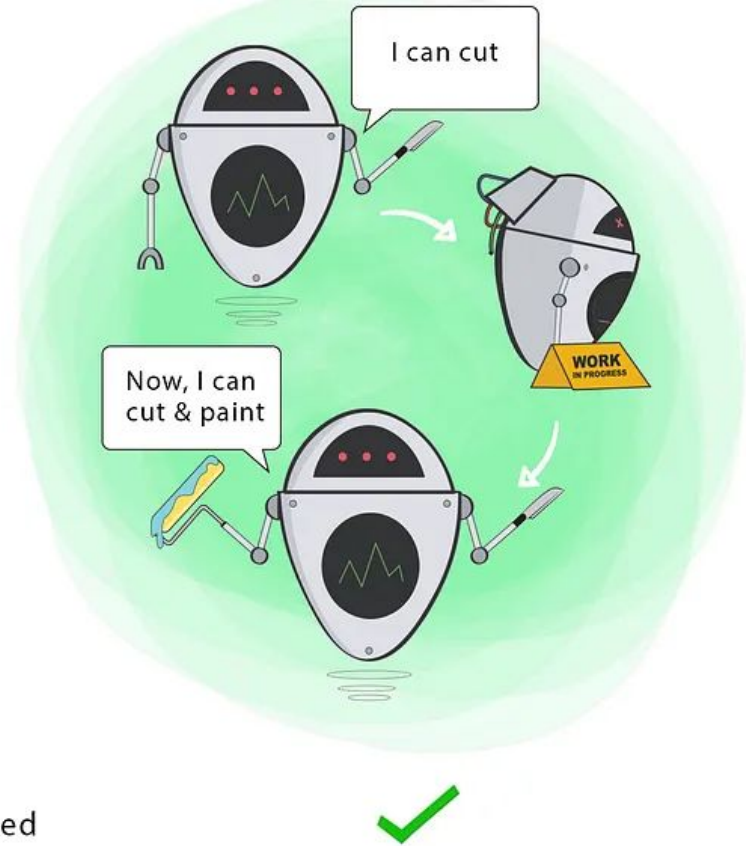
```
public class ProfessorEmTreinamento extends Professor {  
    //code  
}  
  
public class PagamentoEmTreinamentoService extends  
PagamentoService {  
    public void pagar(ProfessorEmTreinamento professor) {  
        //code  
    }  
}
```

SOLID - Open-Closed

Programação Orientada a Objeto II



Open-Closed



Obrigada